Knowledge Networks for Smart World Infrastructures

[1]M. Mulvenna, M. Baumgarten, C. Nugent, K. Curran,
[2]F. Zambonelli, N. Bicocchi, M. Mamei

[1] *School of Computing and Mathematics, University of Ulster, Newtownabbey, BT37 0QB, UK*
{md.mulvenna, m.baumgarten, cd.nugent, kj.curran}@ulster.ac.uk
[2] *DISMI - Università di Modena e Reggio Emilia, Italy*
{franco.zambonelli, bicocchi.nicola, marco.mamei}@unimore.it

Abstract

Current society is witnessing an age of computing ubiquity where the digital world is not longer limited to closed work, home or social environments but increasingly envelops every aspects of private and social life and their surroundings. However, if computing power is to serve us, and the converse is to be denied, then individual components and their rich panoply of services must be able to operate without significant intrusion. To achieve this, such services would require a high degree of supporting knowledge, including knowledge about the social, computational, and physical environments in which they are situated, as well as self-knowledge about their own functioning. While this provides the knowledge with which they can, eventually, manage and configure themselves it does also makes them more self-aware or in short it makes them smarter. However, in order to get 'smarter', the environment, its entities and services need some form of properly represented, well correlated and widely accessible repositories, which leads to the concept of knowledge networks which is the focus of this work.

Keywords: Knowledge Networks, Smart Environments, …

1 Introduction

Future implementations of smart environments will require composite, highly distributed, pervasive services in a situated and fully autonomic way. In other words, they will be made up of components capable of [7, 15] understanding the general and specific context they operate in, that is – physical, technological, social, user, task and request specific contexts – in order to orchestrate their activities towards individual contexts, so as to provide a range of specialized context driven services that are simply not possible or impractical in today's smart environments.

In particular, such services need to be able to improve interactions with the physical and virtual world by providing information about the surrounding environment, again physical and virtual, and exploiting such information to adapt their behavior accordingly, e.g., consider a car navigation system adapting its behavior not only on current traffic conditions but also on current weather, road and car conditions. There is also a requirement for them to be able to get the best of the network infrastructure and resources upon which they operate, being able to ensure sufficient quality of service adaptivity and independently of the actual network characteristics, e.g., independently of the fact that they are required from a Wi-Fi PDA, from a GPRS phone, or from whatever connectivity and connected devices will be available at that time [9]. Services must also facilitate social interactions, by properly reflecting and exploiting the social context in which such services are invoked, e.g., for mere entertainment, or socialization, or in the context of business activities. Finally there have to be able to spontaneously aggregate with each other in order to share and exchange knowledge and / or to support a common goal in a more global and collective fashion.

A central challenge for the above vision to become real is the promotion of suitable solutions for enabling the components of a smart world infrastructure to become context- and therefore situation-aware. Assuming that mechanisms exists to produce all necessary "situational" knowledge (e.g., intelligent sensors, advanced monitoring mechanism, task-, user- and social-profilers) for individual components to exploit such knowledge properly it is necessary that the available knowledge (which could be a dramatic amount) is organized for utilization in an efficient and decentralized fashion. Facilitating the efficient retrieval and understanding of individual situational information implies that any relations between such information is properly represented and correlated according to well-defined ontological constructs. To promote accessibility, it is necessary that information produced locally at one place is properly diffused into networked knowledge repositories whenever this may be of a more global relevance. Also, it may be important that such information can be exploited for mediated (i.e., stigmergic) interactions among the components of such a repository, so as to promote both robust self-organizing behaviors [5] and fruitful cross-layer interactions as required to efficiently access, populate and organize globally distributed knowledge. This leads to the general concept of *knowledge networks* in which all

information about individual context' are properly represented, organized, and correlated, and around which semantically-enriched interactions among individual components of and across smart environments can take place [11].

This paper aims at unfolding the idea of knowledge networks to be used for smart world infrastructures and it is organized as follows. Section 2 outlines the need for such network and positioning the concept of knowledge networks in a smart world infrastructure and discusses related work in the area. Subsequent sections explore the requirements as well as relevant building blocks for individual aspects of a knowledge network. Section 5 elaborates on the semantic and spatial organization of knowledge before Section 6 concludes.

2 Smart World Infrastructures

The area of pervasive computing considers that our everyday environments will be soon densely populated by sensors and actuators embedded in everywhere and in every object, and that most of us will carry or wear some sorts of mobile or integrated device that allow communication with each other and with the environment. In this context, we will be given the possibility of both contextualizing our computing and social activities to the characteristics of the environment and, vice versa, of having the environment adapt to our own needs. Thus, the concept of situation-awareness is a very central one, which translates in the need of properly representing situations occurring in the context. Not surprisingly, several proposals [do we have some references here?] in the area suggests a proper structuring of the available distributed contextual information into sorts of knowledge networks to facilitate the enforcement of contextual activities.

Several everyday activities of humans in an environment relates to orienteering themselves in such environment, e.g., to find places, objects, persons, or source of events, as well as to coordinate movements with other persons. Consequently it may be important for contextual information (generated by an active environment) to be properly structured into semantic or spatial models: the contextual information related to any item/event localized in the environment should be properly propagated and made available, so as to facilitate, from everywhere, the localization of such item/event. Possibly even more interesting, is the possibility of adaptive interactions between users, services or even robots in an active environment.

As an additional considerations, also in the area of pervasive computing – similarly to the area of sensor network – proposals exists [again, a reference would be nice] that suggest exploiting the available contextual knowledge about specific situations to somehow build a prediction of the situations that are most likely to occur in the environment [RoyRD06]. In these cases, the key goal is to predict activities of users in the environment and shape the environment to better accommodate such activities (e.g., by switching on the heating in the living room some minutes before the user will enter it to watch the television).

Most researches on sensor networks focuses on the definition of algorithms [Est02] and, more recently, on programming abstractions [MotP06], to retrieve in an effective way data collected from dispersed sensors and to properly activate sensors on the basis of the sensed data (e.g., to follow a sensed mobile target). In other words sensor networks must by definition act in a context-aware ways: what they do and the way they collect and forward data has to be based on what is happening in the environment they are devoted to sense.

A typical basic exploitation sensor networks is that in which a set of sensors distributed in an environment have to report back to some base station global information about some perceived distributed information (e.g., a temperature or a humidity field). However, in many practical cases, it may be unfeasible for each and every sensor to communicate its locally sensed data directly to a central point. And even if it were, the high communication and energy costs involved in the process of having sensors forward the data may be highly inefficient. Thus, mechanisms have to be implemented that enable sensors to somehow aggregate in a distributed way the sensed data before communicating them to a requesting service.

For instance, service may be conceived that calculates the average value of a localized sensor array, e.g. temperature readings of a certain geographical area.

Some research has been done in this area to provide middleware services that enable, via simple high-level abstractions or APIs, to express the level of abstraction that one wish to obtain from a sensor network [Cur05, Abd04].

What is of interest here is to emphasize that, to some extent, the process of aggregating the data sensed by a sensor network can be assimilated to the process of dynamically building a simple hierarchical structure in which distributed contextual data is pruned, aggregated and preprocessed to be made accessible in a easy, efficient and information rich way.

Another interesting issue in sensor networks is that of adapting a sensor (or any numbers thereof) based on currently sensed information. For instance, a sensor array for motion tracking could adapt its focus based on the past movement of an object as sensed by other parts of the sensor array. In this context, a promising approach (as enforced in the directed diffusion algorithm [IntGE00, Est02]) is that of dynamically build a sort of gradient overlay over the sensor network such that, following the gradient of this overlay, the sensor of interest can be reached. Of course, in the case of mobile targets, the overlay must be dynamically updated to reflect the new situation (e.g., the new sensor that senses the target).

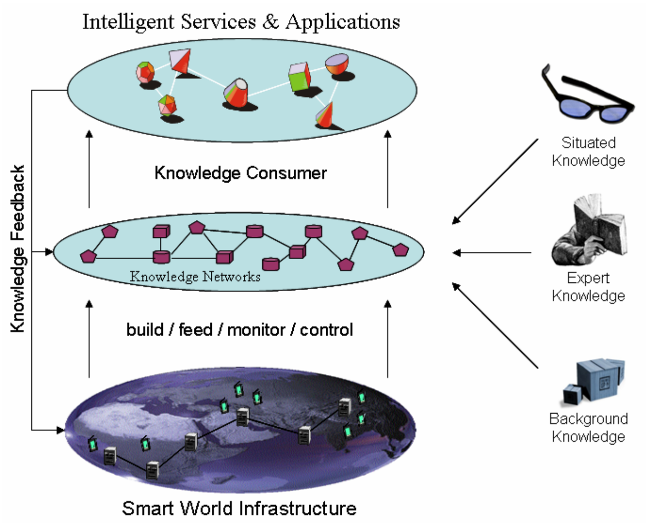

Figure 1: Positioning Knowledge Networks

In the vision of pervasive computing and ubiquities environments, knowledge will be provided through smart sensors which make up local pervasive spaces which in turn could make up a what we would like to call a smart world infrastructure in which sensor based information are available and accessible via a continues network.

As depicted in Figure 1, a dedicated middle layer or *knowledge network* can be introduced to connect the smart world infrastructure with its virtually infinitive number of sensors and devices to a conceptually higher located knowledge consumer layer. Simplified a knowledge networks could be seen as reflective spaces for distributed contextual information. That is, knowledge networks not just intend to store information directly but instead act as a lightweight overlay network upon existing repositories and sensor arrays. However, also being capable of storing distributed, heterogeneous, dynamically constructed and sophisticated knowledge, they can form a conceptual middle layer in which individual physical- as well as application-based components can access information and can coordinate with each other.

Furthermore, future knowledge provisioning systems should be endowed with self-management capabilities but they should, at the same time, meet one of the crucial requirements of ubiquitous and pervasive computing applications: lightness. That is that knowledge should be stored locally where it is collected rather that being stored in a centralized fashion. It also means that knowledge should be distributed by referencing it rather than by duplication. Taking this into account the concept of knowledge networks could provide a mechanism that is (a) flexible: because knowledge can be introduced, removed and maintained in a dynamic fashion and (b) lightweight: because knowledge is kept locally and as such enormous and unmanageable heavyweight knowledge repositories will be avoided. As such the rational of knowledge networks can be synthesized as follows.

Firstly, there is a basic need for expressive and flexible means to promote context-awareness of services and applications. Future smart environments, their components and services need to have awareness of situations with differing degrees of granularity [14]. There is a requirement for some form of computational model of context processing as in [1] that orchestrates context stimuli and components in a coherent representation. Additionally, some way to gauge the quality of contextual information objectively as it is gathered, as from the Quality of Context mechanism of Buchholz *et al.* [2], in which any contextual information comes associated with parameters including precision of information, correctness probability, trust worthiness, resolution and regency. Simply said, contextual information cannot reduce to a trivial set of data to be accessed by components, but requires some higher-form of organization.

Secondly, contextual information cannot be simply considered as local and locally available to components and services. For a satisfactory adaptive orchestration of distributed activities (whether this is intended to be the orchestrated configuration of individual components or the coordination of distributed service components), the exploitation of local knowledge only may not be enough. Nor can one think of concentrating in a single site or of replicating anywhere all available knowledge, especially when this knowledge represents dynamically evolving situations, i.e., it is subject to obsolescence. The compromise solution is to enable components which need more than simply local knowledge to organize and correlate distributed knowledge into sorts of networks that enable distributed components to “navigate” through the available knowledge to attain, on demand, the required degree of contextual awareness.

Third, there is a recognized need for future autonomic communication scenarios to promote cross-layer interactions [13], which is particular relevant for smart environments. This means that the service level and the network level cannot work as separated universes, each towards its own goals. Rather, a continuous exchange of information must occur between the service and the network level, and vice-versa, so as to ensure that the overall activities of the system, at each level, will contribute towards the achievement of a satisfactory functioning. For this coordination and exchange of information to occur without significant interoperability issues, there must be some place where common information and valuable knowledge can be stored and can be properly organized so as to be accessible and understandable by both the network and the application levels.

Fourthly, it is known that a reasonable and effective way to promote self-organization and self-adaptation, which are essential for autonomic behavior, in distributed systems is via stigmergy, i.e., by indirect interactions occurring via a computational environment in which components can spread and sense information [11]. The presence of a distributed

network of knowledge, to be accessed for sensing and effecting by both network and application level components, can act as the computational environment to enforce stigmergic self-organization. Moreover, if such space can contain properly represented and correlated situational knowledge instead of simple digital pheromones, one can think at leveraging stigmergy to more sophisticated forms of cognitive self-organization.

3 Knowledge Networks

Knowledge networks may be conceived as a conceptual layer, positioned between the physical layer (reified in the forms of the environmental information that can be produced by e.g., sensors) and the social layer (reified in the form of social information produced by e.g., social- user- or task-specific experts or applications). In general, the knowledge generated by both the physical and the social layer reaches the same conceptual knowledge level where it is properly put in context of extant knowledge. Furthermore, the knowledge network layer itself may generate additional knowledge about the knowledge it holds or its two adjacent 'worlds' and as such could provide a certain degree of introspection

The presence of specific computational entities in charge of maintaining and updating knowledge networks [CPR+03] is currently not considered, simply because of the fact that such a solution would be too 'heavyweight' and, more importantly, would introduce additional complexities. Instead, it is assumed that components at all layers will be directly in charge of populating, storing, and maintaining portions of fully distributed knowledge networks, arguably with the help of reactive code fragments that are associated to distinct knowledge pieces and aimed at automating their update and maintenance upon changing conditions. While this not only reduces computational efforts orientated towards the maintenance of knowledge networks it will also improve the quality of knowledge because the same group of applications and services that request knowledge from the knowledge network are also in charge of maintaining and organizing the knowledge in the first place.

For instance, assuming that each entity in the knowledge network, whether a software agent or a network device, has the capability of accessing the knowledge network layer for reading the knowledge in it, understanding the relations between individual pieces of knowledge and navigating the links thereof. By this, components can properly understand where newly produced knowledge can be inserted in a knowledge network, and how this has to relate with existing knowledge.

For the actual production and update of the network of relations within a knowledge network, ontological constructs may be used at different levels of granularity of the knowledge network that are constantly adapted via intelligent introspective methods. This behavioral feedback loop, in essence, is the knowledge generator that could dynamically populate required ontological constructs. Such constructs must be designed to be very flexible, even to the extent of facilitating self-revision [6] and they also have to be capable of fusing contexts [8] and knowledge from different ontologies as networks and devices may interconnect in an ad-hoc manner.

Obviously, the construction of a single knowledge network capable of mirroring the 'universe' is illusionary, specifically when considering that even relatively small infrastructures could generate enormous amount of knowledge. A natural extension to the above concepts is therefore the possibility of a multiplicity of knowledge networks to co-exist within a globally accessible knowledge space where each network is limited by clearly defined knowledge boundaries in order to serve application and service specific goals. Such a "network of networks" is conceptually similar to the concept of super-peer networks as provided by e.g. JXTA [SUN05], where each node of a network could represent a cluster of other nodes. However, the need for achieving effective cross-layer interactions and globally coherent activities may require different knowledge networks to be somehow related or interlinked with each other. In particular the possibility of enforcing the construction of scale-free networks of networks, exhibiting a self-similar structure that can facilitate robust navigation and update procedures implies challenging research directions that require the identification of conceptually easy, practical, and scalable ways by which to compose and relate a variety of diverse knowledge networks and the diverse knowledge they contain. Another interesting aspect is the nesting of various knowledge networks into each other, which could enable an exploitation of knowledge networks at different scales (zooming in and out depending on specific needs).

3.1 Requirements

Simplified, a knowledge network is a concept that provides a vehicle capable of gathering, representing, organizing and providing knowledge at different levels of granularity to individual services and applications. In order to provide such a service in an autonomous operating fashion, knowledge networks in general and any building blocks thereof have to support a number of characteristics which are outlined in the following paragraphs. As depicted in Figure 2, conceptual components required for the realization of knowledge networks are threefold.

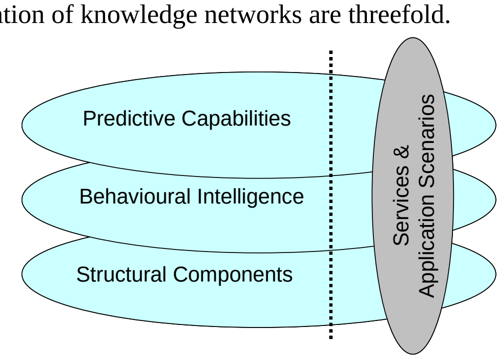

Figure 2: Evolution of Knowledge Networks

Firstly, structural components that provide necessary entities capable of holding knowledge at different levels of granularity including the implementation of a highly flexible framework capable of linking individual knowledge components or any group thereof into distinct purpose-build sub-networks. Secondly, behavioral components which deal with more dynamic aspects of knowledge networks such as knowledge-organization, -optimization, -adaptation and -configuration activities. Thirdly, predictive capabilities enabling detailed analytics of individual knowledge components in order to analyze existing knowledge, reason about the quality of knowledge and to derive new knowledge.

3.1.1 Structural Requirements

The structural requirements laid out in this section provide the conceptual foundation of the knowledge network architecture. In order to construct an efficient and highly flexible framework capable to encapsulate and distribute knowledge on different levels of granularity the following requirements have to be taken into account:

- **Component Based Architecture:** A highly decomposed architecture is desirable in order to increase flexibility and to foster the concept of self-similarity among knowledge entities.
- **Hierarchical Organization of Knowledge:** Distinct hierarchical relations can be maintained or drawn upon individual contexts from which knowledge is collected from and / or provisioned to. The dynamic implementation of such hierarchical constructs is paramount for the concept of knowledge networks in order to enable ad-hoc construction of knowledge structures and for constant and intelligent maintenance of relations among individual knowledge entities.
- **Platform Independent Representation:** To be used by multitude of services and applications within different environments a flexible and platform independence representation of knowledge is indispensable.
- **Dynamic Knowledge Overlay Structures**: Efficient identification, access and referencing mechanisms are required in order to provide knowledge efficiently across physical network structures and to construct purpose build views over distinct areas of interest.
- **Reference based Knowledge Access:** In order to provide high quality, up to date knowledge and to foster the lightweight approach mentioned earlier, knowledge should be requested when needed rather to be stored at certain intervals. Therefore, knowledge entities should reference to the knowledge source instead of storing the knowledge object directly. Obviously, necessary access methods have to be provided as well.

3.1.2 Behavioral Requirements

For the scope of this work the term behavioral intelligence is used to refer to two concepts relevant for the usage of knowledge networks. Firstly, it relates to concepts that deal entirely with the organization of knowledge including the aggregation of smaller knowledge components into higher more meaningful concepts, general lifetime management, quality control mechanism and access optimization. Secondly, it relates to aspects that would make it ideally intelligent enough to operate autonomously. Practically, both concepts are related to each other in a way that individual methods and algorithms used to organize knowledge should ideally operate in an autonomous fashion with no or very little user intervention.

For knowledge networks to become highly autonomous they have to be alert, self-aware and independent, where alert means a high degree of readiness for any kind of spontaneous behavior (action as well as reaction). A high degree of self-awareness is required to enable intelligent behavior and adaptation. Finally, they have to be independent in a way that they operate with no or at least very little human interaction implementing special means of supervision and for advanced contingency planning in the case of emergencies.

A primary need in this direction is to build the knowledge network itself in a self-organized way. The following principles should then readily be applied.

- **Self-organization:** Intelligent devices as well as contextual information and resources are distributed across heterogeneous networks and may be added or removed from a network like structure in an ad hoc manner. In order for services to make use of this distributed information and resources, they must be structured or referenced in an easy-to-access-and-retrieve structure in an automatic fashion. Such contextual information and resources must be autonomously recognized and organized. The autonomous structuring of contextual information and resources is the essential work of self-organization but may work in tandem with self-optimization techniques in order to provide efficient organizational structures.
- **Self-contextualization:** Is the ability of a system to describe, use and adapt its behaviors to its current context. In the case of knowledge networks, this is relates to the adaptation of its internal structures, resources and information.

These characteristics make our proposal clearly distinguished from the "Knowledge Plane" approach [Cla03]. The "Knowledge Plane" is considered as an additional network layer between the network and the application layer, and it is the place in which nearly all network control activities take place. The knowledge plane

as commonly postulated is populated by heavyweight agents, managing and exchanging knowledge about the current state of the network, resulting in directly enactable forms of control over both network and application components. In our experiments, networks of knowledge are not intended to be handled and managed by external entities, but by ACEs themselves.

In our opinion, our view on knowledge network is closer to the concept of overlay networks in P2P computing [AndS04, BabMM02, Rat02, RowD01]. Indeed, autonomic knowledge networks will be sorts of overlay. However – unlike traditional overlay approaches in P2P computing – they are not intended to simply support navigation of data and messages in a dynamic network of components. Rather they are intended to provide application components with a local representation of the situation, that can then be used by them to adapt their behavior e.g. to enforce self-* and autonomic properties.

Swarm intelligence approaches consider that global self-organizing and self-adapting behavior can be made to emerge in systems of a large number of lightweight agents that indirectly interact via the mediation of an environment [Par97, BonDT99, ParBS04]. Agents, by depositing and by sensing "pheromones", and by having the environment properly diffuse pheromones according to specific laws, can – to most extent unconsciously – self-organize their global activities into robust and adaptive patterns.

Our concept of autonomic network knowledge can leverage the traditional concept of stigmergy into a concept of "cognitive" stigmergy. Activities can be driven not simply by reacting to a local concentration of meaningless pheromones, but can be driven by the actual knowledge represented by the network of knowledge. Thus, without requiring ants to become heavyweight agents, an autonomic knowledge network can be enabled to promote more informed "semantic" forms of self-organization. Similar considerations can be made for those approaches to self-organization based on indirect interactions such as the morphogen gradients of "Amorphous Computing" [Nag02, NagM04] and the field-diffusion in teams of mobile robots [MclS04].

Several modern middleware proposals for mobile and ubiquitous computing consider exploiting sorts of distributed data structures – to be dynamically built and self-adapting – to act as the basic mean via which adaptive coordination activities can be promoted. Such middleware proposals include among the others LIME [PicMR01] and TOTA [MamZ04], Smart Messages [Bor04], Limbo [Dav02]. These approaches, by having distributed data structures typically represent some application-level knowledge – other than simply a value to which to react (as is the case of swarm intelligence systems) – definitely shares something with our "autonomic knowledge network approach". However, so far, little has been said on the possibility of building scalable global distributed data structures and on the possibility of exploiting similar sorts of middleware-level data structures to promote self-organization and self-adaptation. However, so far, little has been said on the possibility of building scalable global distributed data structures to instantiate autonomic network knowledge and on the possibility of exploiting similar sorts of middleware-level data structures to promote self-organization and self-adaptation.

In addition to the above characteristics, a number of important self-* and autonomic behaviors are needed to add robustness and flexibility to the knowledge network.

- **Self-optimization:** Distributed contextual information and resources and their availability are rapidly changing. There is a need for autonomous methods that enable consistent monitoring and control of contextual information and resources, so that service/software components may be executed or deployed in the most optimized fashion. Autonomic systems and smart environments must seek to improve their operation over time. They must identify opportunities to make themselves more efficient from the point of view of strategic high-level policies, such as performance, cost or accuracy.
- **Self-configuration/self-adaptation:** Enables autonomous structuring of contextual information and resources thus making them available to services. User services and the underlying supporting services must be re-configured in order to make use of new context information and resources. Newly implemented context information and resources may also trigger changes such as re-configuration within the scope of the knowledge network. Self-configuration and adaptation is therefore a highly desirable aspect that is required on different levels of granularity.
- **Self-healing:** Autonomic systems need to detect, diagnose and repair problems caused by any kind of network or system failures. Using knowledge about the system configuration, a problem-diagnosis embedded intelligence needs to analyze the monitored information. Then, the network may use its own diagnosis mechanism to identify and enforce solutions or alert a human in the case of no solutions being available.
- **Self-protection:** Clearly, there is a need for autonomic systems for self-protection. They must be able to defend themselves as a whole by reacting to, or anticipating, large-scale correlated problems arising from malicious attacks or cascading failures that remain uncorrected by self-healing measures. For that to work, efficient failsafe contingency procedures need to be implemented.
- **Self-programmability:** Is the ability of a system to "re-program" itself or any other relevant component in order

to adapt for changing conditions. Such behavior could be seen as the ultimate step for autonomic computing because of the fact that adaptation of a system may be achieved by re-programming it.

For truly usable knowledge networks to become reality, all of the above features are required to operate without, or at least with very little, intervention by human.

3.1.3 Predictive Requirements

After the structural building blocks and the behavioral component coordination, predictive capabilities may be seen as the next step in the evolution of knowledge-based systems, such as the knowledge network approach. For such systems, the ultimate goal can be summarized as the provision of accurate, real-time predictions of any kind about individual objects, entities, relations or higher concepts that are embraced by the knowledge provisioning system or that are required by a service utilizing the system. Conceptually, high level requirements for predictive knowledge networks are twofold and can be summarized as the ability of a system to learn and reason about itself, the data and the concepts it incorporates.

In [CPR+03], learning is described as the principled accumulation of knowledge over time. Independent of the type, format and method used learning can take place in various ways such as by observation, by instruction, by summarization, by generalization and others. Reasoning on the other hand can be seen as pre-requisite for predictions which utilizes the knowledge derived through learning mechanism in order to generate new inferences of beliefs [CPR+03]. Reasoning can also be achieved through various mechanisms ranging from traditional statistical methods to advanced semi-automated knowledge discovery methods such as the discovery of rules, behavioral and operational patterns, classifications etc. Furthermore, knowledge derived through reasoning may also be used to achieve introspective learning capabilities, where such knowledge is used to optimize the system it has been derived from.

The extraction of associative and sequential based patterns on knowledge that is available at different levels of granularity is of particular interest as such patterns can be used to identify diverse kinds of relations and for the prediction of short as well as long term behavioral patterns, which may be used to identify access violations, predict the use / disuse of services, optimize knowledge access patterns, etc.

In essence, two types of predictive features are seen to be of particular interest for the scope of knowledge networks. Firstly, forecasting capabilities for sensor based information in order to predict specific context changes before they actually occur. Among other features, this would allow the implementation of pre-emptive measures to e.g. correct system behavior, identify dangerous (or interesting) developments or to simply forecast, for example, network traffic or weather information. Secondly, the extraction and re-integration of so called behavioral and operational patterns, which relate to the registering, updating and usage of knowledge within the scope of a more global oriented knowledge network. In order to provide predictive capabilities for a system such as envisioned by knowledge networks two key types of services have to be provided by the system. These are the concepts of history tracking and event matching and correlation.

Among a host of statistical methods and AI based algorithms the concept of history tracking has to be implemented into the structures of the knowledge network in order to enable predictive capabilities. History tracking relates to functionality that allows effective observation and storage of any kind of entities that are suitable for prediction, such entities may include sensor based information, events, access patterns, etc. However, unlike traditional tracking systems, such functionality has to be built directly into the components of a knowledge network. That is the component that holds the knowledge should also track the history thereof. On a wider scale, that is if applied to the whole knowledge network or to purpose build sub-networks, such functionality resembles the concept of a virtual context warehouse where past contexts can be stored and accessed. A number of flexible methods may be built in into the context warehouse in order to provide certain features such as aggregate values, summarizations, etc. Furthermore, advanced algorithms may be used in real or batch mode to extract new and more sophisticated knowledge which may be introduced into other parts of the knowledge network.

Utilizing the observation functionality of the above, the role for the event matching and correlation service is to efficiently observe, correlate and match incoming events in order to identify interesting behavior of a system and if necessary react on it. If applied to system properties, this will provide functionality for self-supervision, which has been identified earlier on as one of the requirements to achieve autonomous behavior.

3.2 Components

Based on the above assumptions and observations it becomes clear that a new type of knowledge storage and provisioning vehicle is needed. One that provides well-structured and dynamic request-based views to sensor-based knowledge that is directly usable and accessible to individual services and applications. Additionally, such a vehicle must be lightweight; in order to avoid large scale data duplication and it must be self-similar to be flexible and scalable enough to accommodate for next generation data sources provided by upcoming smart environments.

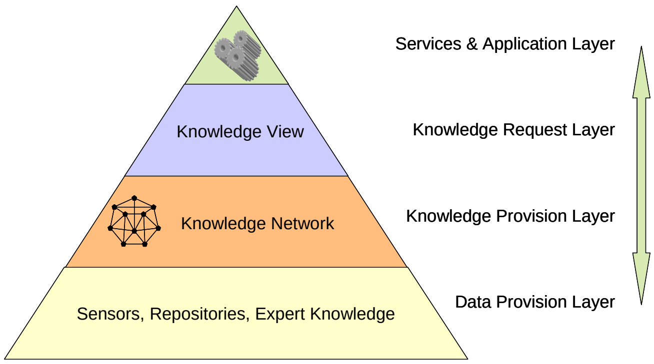

Figure 3: Knowledge Provisioning

As depicted in Figure 3, a knowledge network needs to connect to some sort of data layer that resides, from a knowledge provisioning point of view, below a knowledge network. On the other hand a dedicated knowledge view layer is required to create temporal views of individual parts of knowledge without changing any parts of the knowledge network itself. This is necessary to provide request based and ad hoc created knowledge structures to knowledge "requesters" which are at the top of the knowledge provisioning pyramid. While the former concept requires the implementation of intelligent methods capable to access a multitude of factual and virtual based data sources e.g., sensors, repositories, smart environments, etc., the latter requires advanced and distributed network management facilities that, ideally, are embedded within the structure of the knowledge network itself.

Two other requirements that have to be taken into account are identified by the need for any component and method to operate in a distributed environment and that the constructs that will eventually make up a global knowledge network should have no theoretical boundaries with respect to the amount of knowledge their may embrace or the type of knowledge to be handled.

Finally, individual components must have the capacity to retain and maintain a memory that comprises the data and knowledge sources they embrace as well as relevant information of neighboring components in order to maintain the distributed structures of the network. This 'memory' needs to be a machine-understandable syntax, comprising different standards in order to maintain semantic integrity and coherence of the knowledge embraced.

Conceptually, two types of components are required to realize the above outlined concepts.

Firstly, a component (Knowledge Atom) is required that provides the link to the data provision layer and as such enable access to a range of sensors, devices and repositories. Secondly, designated and dynamically maintainable relations have to be overlaid upon those entities connecting them to a purpose build network based structure. This requires the provision of flexible container component (Knowledge Container) which implements designated functionality to (re-)configure relations of registered knowledge components and as such enables the construction of purpose build collections of knowledge.

3.2.1 Knowledge Atoms

Representing the most basic component of a knowledge network a knowledge atom is depicted in Figure 4, containing a knowledge object and relevant descriptions that provide the context of the object. Simplified, a knowledge atom encapsulates two concepts: firstly it provides the knowledge object which reflects a single knowledge entity independent of its type, size or context; secondly it has relevant descriptions of the knowledge object attached providing relevant context, system and usage based information that are relevant for the creation, maintenance and observation of the knowledge object. For instance, a knowledge atom could encapsulate the reading of a single sensor (e.g. temperature reading @ location GPS_COORDINATES[X]), where attached descriptions could include the GPS location of the sensor, the purpose of the sensor, the required update frequency, etc. On the other hand a knowledge object could reflect a more complex structure such as e.g. the DNA code of a person. Theoretically, there should be no limits to the type, size, or representation of the knowledge referred to by the knowledge object as long as it is properly described through its descriptive object.

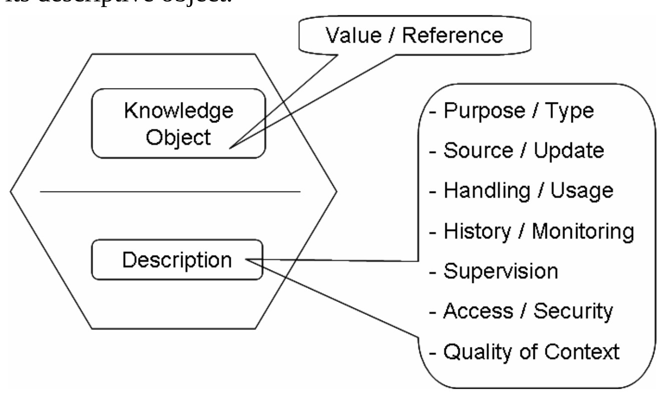

Figure 4: Knowledge Atom

Within the context of a global knowledge network, knowledge atoms may be seen as 'protected' objects, that is that they are not divisible, which is based on the simple fact that, independent of its complexity, they only embrace a single fragment of knowledge. Nonetheless, the description attached to the knowledge object may be extended or manipulated if necessary. It is also envisioned that knowledge atoms and the knowledge they embrace exist locally rather than in a distributed environment. However, as a whole knowledge atoms may be shared, cloned, referenced or transported globally throughout the network.

3.2.2 Knowledge Container, KC

A knowledge container or KC is envisioned as a shell-like structure capable of encapsulating knowledge at different levels of granularity. In fact a KC needs to be capable of encapsulating a knowledge network itself or any part thereof in order to build networks of networks.

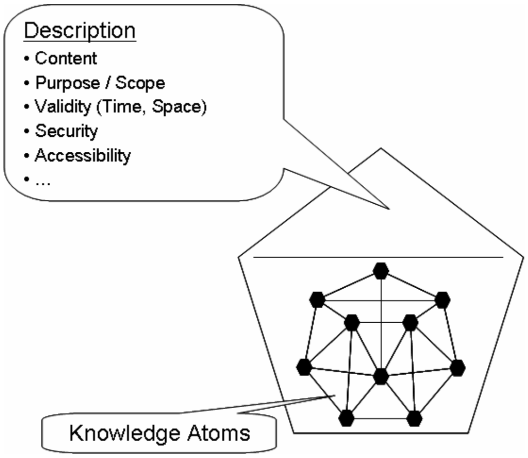

Figure 5: Knowledge Container

As depicted in Figure 5 the underlying concept of a knowledge container is similar to the concept of a knowledge atom. That is that is encapsulates knowledge. However, unlike knowledge atoms the purpose of a knowledge container is to organize knowledge in a semantic or spatial fashion rather than providing access to the underlying information. Basically a knowledge container or KC may embrace (or reference) any number of knowledge atoms independent of their location (locally or remotely). Organizing them hierarchically the rationale of an AC is to provide purpose-built, structured knowledge on a higher level of granularity. Similar to knowledge atoms additional descriptive information may be attached to a KC, thus providing relevant information about the type, scope, purpose, usage, etc. of the knowledge they embrace, the purpose they were created for and the way they are used. Unlike knowledge atoms, KC's are freely extendable, divisible and modifiable, that is, that new knowledge atoms can be added, other may be removed or that hierarchical links between them may be modified, created or removed at any time. Please note that although some concepts represented through the descriptive part of a KC and a knowledge atom respectively may be shared or inherited, they are not the same and therefore not necessarily equivalent.

Extending the example above a knowledge atom a KC could embrace a number of sensor readings (e.g. temperature, wind, humidity etc.) that together reflect a higher concept, in this case weather information @ a specific location.

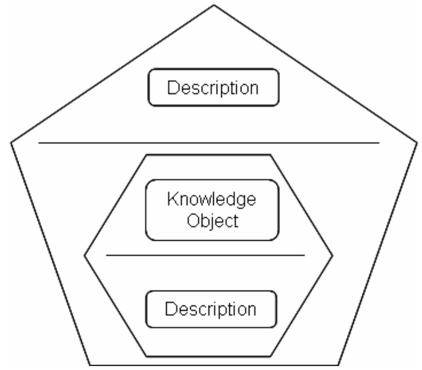

Figure 6: Knowledge Atom Wrapping

In order to enable the construction of larger highly distributed knowledge network structures and to support the concept of self-similarity, individual knowledge atoms are always wrapped by a KC, which is visualized in Figure 6. This allows the construction of knowledge networks utilizing the concept of a KC only. Or in other words, knowledge atoms store the knowledge whereas KC's are used to organize them. This concept also enables the construction of networks of networks where each node (KC) of a network-like structure may contain a network itself which in turn could contain networks and so on. Within a P2P environment this concept is known as the construction of super networks which is depicted through Figure 7.

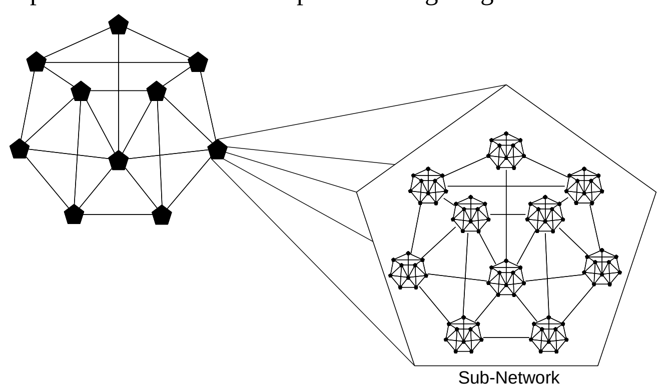

Figure 7: Super Networks

Extending the weather example further the purpose of a smaller knowledge network could be the provision of weather information of all major cities in e.g., the United Kingdom. Assuming that there exist relevant sensors in all cities concerned and that their also exists a dedicated KC for each city (preferably individual KC's reside on a computational resource that is somehow connected to the city to which they belong) then another KC (e.g., WEATHER(United Kingdom)[AKC[0], AKC[1], AKC[2], …]) could be created that embraces all other KC's concerned thus providing a central dedicated knowledge resource where individual KC's can be added or removed automatically and

which can provide relevant information to other services and application.

Alternatively, the KC's embraced by the WEATHER(United Kingdom) KC may be automatically re-grouped by a dedicated service that is available through what we have previously termed behavioral intelligence components. In this example the original KC may be sub-grouped even further taking additional information into account such as discrete geographical regions thus extending the above example as follows:

WEATHER(United Kingdom)
[
WEATHER(England) [KC[0], KC[1], KC[2], …],
WEATHER(Scotland) [KC[0], KC[1], KC[2], …],
WEATHER(Wales) [KC[0], KC[1], KC[2], …],
WEATHER(Northern Ireland) [KC[0], KC[1], KC[2], …]
]

Note that the root element of the knowledge network described above is a knowledge container, which may be used as part of another network in the same way this network is build upon other KC's. Although simplistic, the above weather example nicely validates the suitability of the KC approach and shows its flexibility and extensibility towards more complex scenarios. It also shows that KC's can be expressed formally through a dedicated mark-up language, such as XML, which fosters interoperability among different physical and virtual resources.

4 Vertical vs. Horizontal Organization

The fine-grain topology and structure of the knowledge networks will be mainly application-dependent in that it will be used to inter-relate concept and context-information so as to be useful for a given application task.

Despite of this, at a coarser granularity, we think of two main different kinds of structures to shape these knowledge networks: vertical and horizontal structures.

Vertical structures are those relationships that involve ACE's linked in a sort of protocol stack that enable to transform low-level data into high-level application knowledge. Vertical structures mainly involve ACE's that are linked so that one ACE uses and abstracts towards high-level concepts the data produced by the ACE below (in the protocol stack).

For example, in a sensor network scenario, a number temperature readings could be aggregated together into the higher-level concept of fire outbreak. Given that knowledge network, the application could rely on the "fire" knowledge element disregarding the low-level knowledge on temperature.

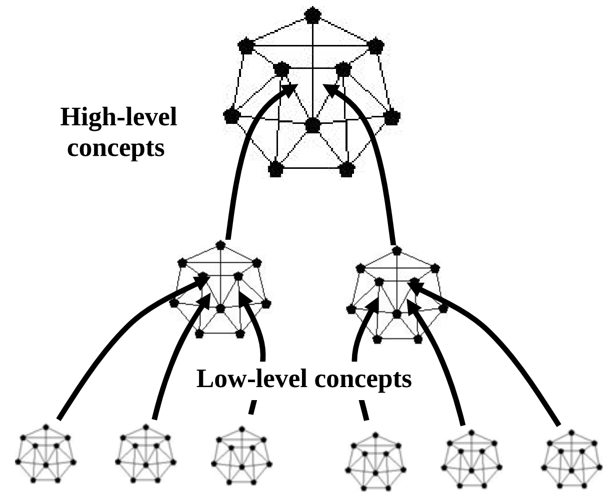

Figure 8: Semantic structure of a knowledge network

Horizontal structures are those relationships that allow ACE's, conceptually located at the same level of abstraction, to aggregate and link together their knowledge networks. Horizontal structures can be used to chose and select an area in the knowledge space on the basis of a given attribute. For example, in a sensor network scenario, horizontal structures could link the nodes reading a temperature within a given range, and could be used – for example -- to get their average values. Horizontal structures are likely to be (spatially) distributed on a wider area than vertical ones. To this end it becomes fundamental to shape these links trying to achieve topologies that offer suitable performance in spreading and collecting information.

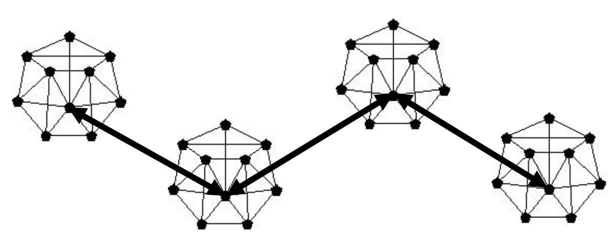

Figure 9: Horizontal structure of a knowledge network

5 Conclusions

Khkjh

G

Fg

Fg

F

Gf

g

6 Acknowledgments

Work supported by the project CASCADAS (IST-027807) funded by the FET Program of the European Commission.

7 References

[1] [SUN05] SUN Microsystems, "JXTA Technology", http://www.sun.com/software/jxta/, March, 2005.

[2] [CPR+03] D. Clark et al., "A Knowledge Plane for the Internet", Proceedings of the 2003 ACM SIGCOMM Conference, Karlsruhe (D), ACM Press, 2003.

[3] [RoyRD06] N. Roy, A. Roy, S. K. Das, "Context-Aware Resource Management in Multi-Inhabitant Smart Homes: A Nash H-Learning Based Approach", 4th IEEE Conference on Pervasive Computing and Communication, Pisa (I), March 2006.

[4] [Est02] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, 1(1):59-69, Jan. 2002.

[5] [MotP06] L. Mottola, G. P. Picco "Programming Wireless Sensor Networks with Logical Neighborhoods", Proceedings of the 1st International Conference on Integrated Internet Ad hoc and Sensor Networks (INTERSENSE 2006), Nice (France), May 29-31, 2006.

[6] [Cur05] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A.L. Murphy, and G.P. Picco, TinyLime: Bridging Mobile and Sensor Networks through Middleware, In Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom'05), IEEE CC Press, pp. 61-72, March 2005.

[7] [Abd04] T. Abdelzaher, et al., "EnviroTrack: Towards an Environmental Computing Paradigm for Distributed Sensor Networks", 24th International Conference on Distributed Computing Systems, IEEE CS Press, Tokyo (JP), May 2004.

[8] [IntGE00] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks",ACM Mobicom, Boston (MA), USA, 2000.

[9] [Est02] D. Estrin, D. Culler, K. Pister, G. Sukjatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive Computing, 1(1):59-69, Jan. 2002.

[10] [6]Heflin, J., Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment, PhD., Thesis, University of Maryland, College Park, 2001.

[11] [8]R. Mayrhofer, H. Radi, A.Ferscha. Recognizing and predicting context by learning from user behavior. 2003: Austrian Computer Society (OCG).

[12] [14]Sterritt, R., M.D. Mulvenna, and A. Lawrynowicz, "Dynamic and Contextualised Behavioural Knowledge in Autonomic Communications", Proc. of the 1st IFIP Workshop on Autonomic Communications. Berlin: Springer-Verlag, 2004.

[13] [1]C. Balkenius, J. Moren, "A Computational Model of Context Processing", 6th International Conference on the Simulation of Adaptive Behaviour. The MIT Press, 2000.

[14] [2]Buchholz, T., A. Kupper, and M. Schiffers, "Quality of context: What it is and why we need it", Workshop of the HP OpenView University Association 2003 (HPOVUA 2003), Geneve (CH), 2003..

[15] [13]EU IST Commission, "Situated and Autonomic Communication Initiative", FET Report, www.cordis.lu/ist/fet/comms.html, 2005.

[16] [11] V. Parunak, "Go to the Ant: Engineering Principles from Natural Agent Systems", Annals of Operations Research, 75:69-101, 1997.

[17] [7] J. Kephart, D. Chess, "The Vision of Autonomic Computing", IEEE Computer, 36(1), 2003.

[18] [15] F. Zambonelli, M.P. Gleizes, R. Tolksdorf, M. Mamei, "Spray Computers: Explorations in Self-organization", Journal of Pervasive and Mobile Computing, 1(1):1-20, May 2005.

[19] [9] M. Mikic-Rakic, N. Medvidovic, "Support for Disconnected Operation via Architectural Self-Reconfiguration", International Conference on Autonomic Computing, New York, NY, USA, 2004.

[20] [5] G. Di Marzo, A. Karageorgos, O. Rana, F. Zambonelli (Eds.), Engineering Self-organizing Systems: Nature Inspired Approaches to Software Engineering, LNCS No. 2977, Springer Verlag, May 2004.

8 Biographical notes

Franco Zambonelli is a professor of computer science at the University of Modena and Reggio Emilia. His research interests include distributed and pervasive computing, multiagent systems, and agent-oriented software engineering. He received his PhD in computer science from the University of Bologna. He is a member of the IEEE, the ACM, AIIA, and TABOO.

Nicola Bicocchi is PhD student at the University of Modena and Reggio Emilia, where he received the Laurea in Computer Science. His current research interests include distributed and pervasive computing, swarm intelligence and

information security. He is a member of the AIIA, and TABOO.

Marco Mamei is an associate researcher at the University of Modena and Reggio Emilia, where he received the PhD in computer science. His current research interests include distributed and pervasive computing, swarm intelligence, and multiagent systems. He is a member of the IEEE, AIIA, and TABOO.

Maurice Mulvenna is a senior lecturer in Computing Science at the University of Ulster's School of Computing and Mathematics in the UK. He has published over 70 papers in the area of artificial intelligence, addressing both theory and practice and is a grant holder in research and venture project worth over €6M. He is a proposal rapporteur, appraiser and project reviewer for the European Commission. As a co-founder and CEO of a research-led start-up company he also has experience in steering basic research to commercialization. He is a member of the BCS, IEEE and ACM.

Matthias Baumgarten is currently working as a research fellow in the Faculty of Informatics at the University of Ulster. His main research interests are in the field of data mining in particular web mining, pattern discovery and personalization techniques. He has received his PhD in 2005 and has (co) authored about 18 publications

Chris Nugent is a Senior Lecturer at the University of Ulster, in Jordanstown. He is an active researcher and a member of numerous journal and conference committees. His research interests include Artificial Intelligence; Medical Informatics; Multivariate Data Analysis and Neural Networks. Within these areas he has published over 120 papers in journals, books and conferences and has recently co-edited a text in Personal Health Management Systems.

Kevin Curran BSc (Hons), DPhil, SMIEEE, MBCS CITP, MACM, MAIAA, MIEE, ILTM is a Lecturer in Computer Science at the University of Ulster. He has published over 170 publications to date in the field of distributed computing especially emerging trends within wireless ad-hoc networks, dynamic protocol stacks and middleware. He is a member of the Editorial Committee of the Journal of Universal Computer Science (J.UCS).